

HOW IT WORKS

Public key cryptography

It's essential that you keep private and sensitive information secure when you're online, but how does it happen? Tom Royal decodes the intricacies of encryption

Computers and the internet have made moving information incredibly straightforward. We're so used to sending an email to the other side of the world in a few seconds, it's easy to forget what a remarkable feat this is. These days, we also trust the internet with our money. It's simple to check your bank account balance or set up a money transfer from your PC, and more and more people are buying everything from groceries to cars online.

But there's a key difference between a regular email and a financial internet transaction. The email can be compared to a postcard. After you send it, the message can be read by anybody who handles the message on its route to the recipient. But if you're placing your weekly organic mango order with *www.ocado.com*, you won't want your bank or credit card details to be so easily accessible. For this reason, financial and other sensitive transactions made online use an encrypted connection.

You can tell that your connection to a website is encrypted because a little padlock appears in the status bar of your web browser. It relies on technology known as public key cryptography. You can use the same technique to secure your emails from prying eyes.

SIMPLE SYMMETRY

The simplest way to encrypt a message is known as symmetric key encryption. Symmetric key encryption uses a key to encrypt a message, and the same key to decode it. One of the earliest and simplest forms of symmetric key encryption

is Caesar's Cipher, where each letter of a message is replaced by another a certain number of letters further along the alphabet. If using an offset of three, the message:

TOP SECRET MESSAGE

is encoded to:

WRS VHFUHW PHVVDJH

Like any other symmetric key encryption, this method requires the receiver, in this case one of Caesar's generals, to hold the same key. When he is told that an offset of three is being used, the message is ridiculously easy to decrypt.

Symmetric key encryption forms the basis of the common Data Encryption Standard (DES) encryption. This system was adopted by the US government in 1976 for the protection of non-classified data, and its use was widespread in other countries for some time. Instead of a simple offset, DES uses a 56-bit number as its key for encryption and decryption of a message.

With 2^{56} (72,058 million million) different possible keys, DES seems secure, but its biggest problem is getting the key to the recipient securely. If we wanted to send an encrypted copy of *Computer Shopper* to a reader in Australia, we'd need to send them the key by some means, too. Symmetric encryption means that anyone who managed to intercept the key would be able to decrypt our message and steal the magazine. Not only that, they could send a

bogus message in its place – perhaps containing a copy of *Nuts* magazine – and the reader would think it had come from us, because the shared key unlocked it.

The consequences of a symmetric key falling into the wrong hands are doubly bad, as you cannot trust that messages are secret or that they were encrypted by the people you thought they were. But how can you send a symmetric key safely? Sending the key to our reader via email would be very unsafe, and a message sent through the post could be intercepted and opened. Sending it with a courier might work, but the courier would have to be someone we trust implicitly. The only really secure way to get the key to our reader would be for one of the editorial staff to convey it personally to Australia.

Unfortunately, even personally delivering our key would not guarantee the security of our message if it were encoded using DES. Though the 56-bit key used by DES sounds impressive, it's not long enough, and advances in computer technology since it was designed have meant that it is now relatively simple to break using brute force. In 1998, the Electronic Freedom Foundation (EFF) used a custom-built computer known as Deep Crack and a distributed computer network to break a DES-encrypted message in less than 24 hours. Today, DES has been largely replaced by Triple DES, which applies encryption three times for extra security, and the Advanced Encryption Standard (AES).

PUBLIC SCHOOL

Sending keys separately via first class post doesn't sound so bad, but it would make many common internet tasks impractical. Buying from an online shop, for example, wouldn't seem so convenient if you had to deliver a key to secure all your transactions. Ideally, you need a type of encryption that doesn't require keys to be exchanged securely. This sounds tricky, but it's possible. The solution is known as Public Key Infrastructure or PKI encryption.

Public key encryption allows you to publish an encryption key for the world to see, and then use it for secure communications. Here's how public key encryption would work with our reader in Australia. First, the reader would need a set of encryption keys. He has two: one for encrypting messages and one for decrypting messages. The former, used for encryption, is known as the public key. This key can be freely distributed because it is capable only of encrypting – not decrypting – messages. The second key, which performs the decryption, is known as the private key. This must be kept securely on our reader's computer and not divulged.

Safe shopping How an SSL connection is made

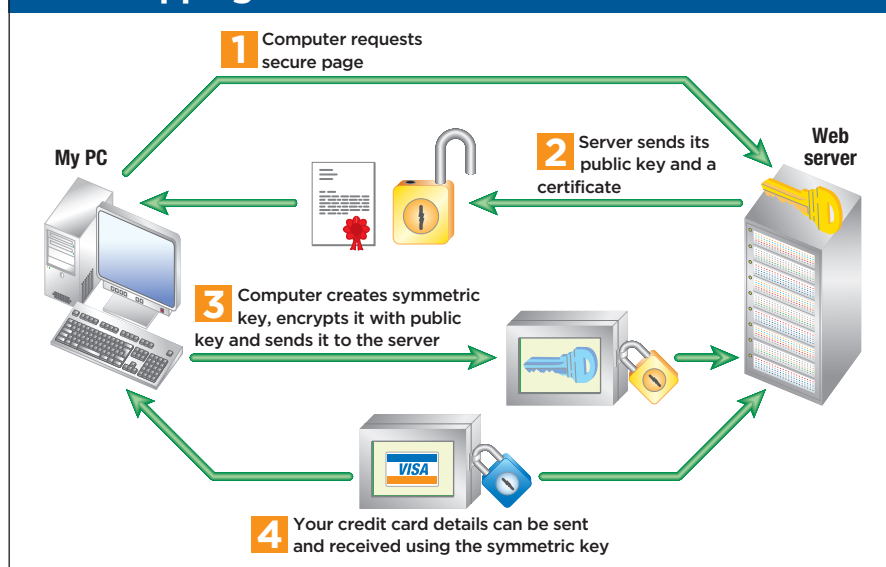


DIAGRAM: MIKE HARDING

Once he has this pair of keys, our reader sends us the public key. As this key can't decrypt, there's no need to go to extravagant lengths to get it to us securely. He can simply email it to us, post it on his website or put it in an online directory for us to look up. Once we have his public key, we encrypt a copy of our message and send it to him. The encrypted message can be decrypted only using the private key. Even if our message is intercepted by someone who knows the public key, they will not be able to read or tamper with it. Only our reader has the private key (stored safely on their computer) so only they can decrypt the message.

PRIME TIME

You may have already spotted the clever bit in all this. How do you create a key that can encrypt but not decrypt? The answer is a mathematical phenomenon known as a trap door. A trap-door function is a mathematical function that is easy to perform in one direction but far harder to reverse. A number of different trap-door functions are in use, but one of the simplest to understand is that used by the popular RSA algorithm, invented in 1977 and named after its inventors: Ron Rivest, Adi Shamir and Len Adleman.

The trap door used by RSA relies on two principles. First, that it is far easier to multiply two numbers together than to start with the answer and work out what the original two numbers were. For instance, with a computer it's easy to multiply two large numbers thus:

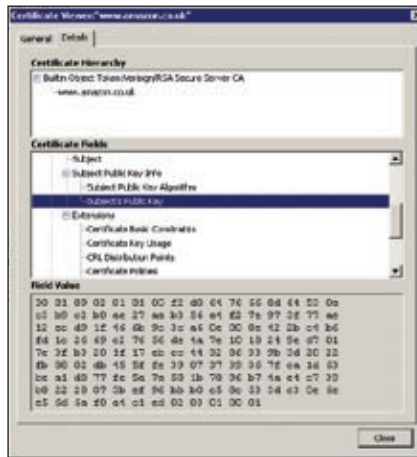
$$2,500,163 \times 3,550,439 = 8,876,676,221,557$$

However, the reverse process, which is called factorisation, is much harder. If you are only given the end result of 8,876,676,221,557, how would you go about finding its factors? In fact, the main technique still used for factorising is trial and error: divide 8,876,676,221,557 by every number you can think of until an integer is produced. Mathematical 'sieve' tricks and computers that process thousands of operations every second make this far faster than it would be by hand, but it's still time-consuming.

The second principle is that any number you choose must be either a prime number (divisible only by itself and one) or the product of a unique set of prime numbers. This is known as the fundamental theorem of arithmetic. The numbers 2,500,163 and 3,550,439, used in the last equation, are prime numbers, and the fundamental theorem of arithmetic tells us there is no other way to factorise 8,876,676,221,557 than as $2,500,163 \times 3,550,439$.

RSA encryption uses these rules to produce a system whereby it's easy to encrypt information, but far harder to decrypt it. To generate the necessary keys, it uses prime numbers. The public and private keys are generated from two very large prime numbers, and both formulae include the product of these two prime numbers.

Because of the fundamental theorem of arithmetic, we know that the public key is uniquely linked to the two prime numbers used to generate it, and therefore uniquely linked or 'paired' to the public key. But we also know that the difficulty of factorisation means that working out the prime numbers used to create the public key is a Herculean task. And without these prime



▲ You can examine the public key sent to you by a secured shopping website and check who issued its certificate

numbers, you won't be able to calculate the private key's value from its paired public key.

POWER IN NUMBERS

The RSA encryption used in online transactions is complicated as it uses long keys for increased security. However, it's easy to demonstrate the maths using smaller numbers. Imagine that you want to create a private and public key so a friend can send you coded messages. We start by choosing two prime numbers, p and q , and multiplying them together to find pq :

$$\begin{aligned} p &= 67 \\ q &= 53 \\ pq &= 3,551 \end{aligned}$$

Next we need to pick a value called e . The value of e must be greater than one, less than pq and relatively prime to $(p-1)(q-1)$. This means that it must have no prime factors in common with $(p-1)(q-1)$, or in this case 3,432. We'll use 19:

$$e = 19$$

This value is known as the public exponent. Next we need to compute a value for d , the private exponent. This number must be such that:

$$(de - 1)$$

is evenly divisible by:

$$(p-1)(q-1)$$

The easiest way to find this number is to enter different integer values of x into the following equation until it produces an integer value for d :

$$d = (x(p-1)(q-1) + 1)/e$$

In our case, this equates to:

$$d = (3432x + 1)/19$$

Entering the value of $x = 49$ gives an integer result of 8,851, so we can use:

$$d = 8,851$$

Our two encryption keys are:

$$\begin{aligned} \text{Public key: } pq &= 3,551, e = 19 \\ \text{Private key: } pq &= 3,551, d = 8,851 \end{aligned}$$

All the other values, including the two original prime numbers, can be safely deleted from the disk and erased from memory. The public key is published online or emailed to our friend. If they now want to send us a message, it must be converted into a numeric value. We'll imagine that this message, m , is 123:

$$m = 123$$

The equation used to encode a message using the public key is:

$$\text{encoded message} = m^e \bmod pq$$

Mod means divide the first number by the second number and keep only the remainder. For example, $106 \bmod 10$ equals 6. Using the values in the public key, we can calculate:

$$\begin{aligned} \text{encoded message} &= 123^{19} \bmod 3,551 \\ &= 1,440 \end{aligned}$$

This message can now be sent to us. To decode it, we apply the following formula using contents of our private key:

$$\begin{aligned} \text{decoded message} &= m^d \bmod pq \\ &= 1,440^{8,851} \bmod 3,551 \\ &= 123 \end{aligned}$$

This particular encryption is not sufficiently secure as the numbers used are very small. RSA Security currently recommends the use of 1,024-bit keys for data that needs to be secure up until the year 2010. A 1,024-bit key is equivalent to a decimal number of over 300 digits.

SECURE SHOPPING

When you access a website that's secured by SSL, the server uses both public and symmetric key cryptography to ensure that the data you send, including your credit card details, is safe from prying eyes. Initially, the server sends its public key to your browser. This is accompanied by a certificate that verifies the website's identity and includes the details of the organisation that issued the certificate. This is usually a company such as Thawte or VeriSign that specialises in issuing website security certificates. If your browser trusts that certificate authority, it will generate a random key, encrypt it using the web server's public key and send it back to the site.

Once the site has decrypted this message (using its private key), the details of your online transaction are secured using the random key for symmetric encryption. This random key is used only for that single session. In effect, SSL uses public key encryption to solve the problem of how to communicate symmetric keys by sending it safely using public key encryption.

You can also secure your email messages using public key encryption. One of the best-known encryption software packages is PGP, available from www.pgp.com. This software now costs around £72 including VAT, but a limited free version is available. Alternatively, the GNU Privacy Guard (www.gnupg.org) is available to download for free. 